

CLAIMS

1. A method for ascertaining application program interface (API) coverage for a Java 2 platform Enterprise Edition (J2EE) application, comprising the operations of:

analyzing an API of an application to determine a plurality of methods comprising

5 the API;

exercising the API utilizing a test program that calls methods from the plurality of methods comprising the API;

generating a list of methods called by the test program; and

comparing the list of methods called by the test program with the plurality of

10 methods comprising the API.

2. A method as recited in claim 1, further comprising the operation of determining a percentage of the plurality of methods comprising the API that are called by the test program.

15

3. A method as recited in claim 2, further comprising the operation of designating the application as portable when the percentage is greater than a predetermined bar percentage.

4. A method as recited in claim 3, further comprising the operation of designating the application as non-portable when the percentage is less than the predetermined bar percentage.

5. A method as recited in claim 1, further comprising the operation of intercepting method calls from the test program utilizing stubs for corresponding methods of the plurality of methods comprising the API.

6. A method as recited in claim 5, further comprising the operation of recording the intercepted method calls.

7. A method as recited in claim 6, further comprising the operation of passing the intercepted method calls to corresponding methods of the plurality of methods comprising the API.

8. A computer program embodied on a computer readable medium for ascertaining application program interface (API) coverage for a Java 2 platform Enterprise Edition (J2EE) application, comprising:

a code segment that analyzes an API for an application to determine a plurality of methods comprising the API;

a code segment that intercepts method calls utilizing stubs for corresponding methods of the plurality of methods comprising the API;

a code segment that generates list of intercepted methods calls; and

a code segment that compares the list of intercepted methods calls with the plurality of methods comprising the API.

9. A computer program as recited in claim 8, wherein the method calls are generated by a test program that exercises the application API by calling methods from the plurality of methods comprising the API.

10. A computer program as recited in claim 9, further comprising a code segment that determines a percentage of the plurality of methods comprising the API that are called by the test program.

11. A computer program as recited in claim 10, wherein the application is designated as portable when the percentage is greater than a predetermined bar percentage.

12. A computer program as recited in claim 11, wherein the application is designated as non-portable when the percentage is less than the predetermined bar percentage.

5 13. A computer program as recited in claim 8, further comprising a code segment that passes the intercepted method calls to corresponding methods of the plurality of methods comprising the API.

10 14. A computer program as recited in claim 8, further comprising a code segment that reads a deployment descriptor for the application to determine a public interface for the application.

15 15. A computer program as recited in claim 14, further comprising a code segment that introspects class Java code for the public interface to generate a list of the plurality of methods comprising the API.

16. A system for ascertaining application program interface (API) coverage for a Java 2 platform Enterprise Edition (J2EE) application, comprising:

logic that analyzes an API for an application to determine a plurality of methods
20 comprising the API;

logic that generates a list of methods called by a test program; and

logic that compares the list of methods called by the test program with the plurality of methods comprising the API to determine a percentage of the plurality of methods comprising the API that are called by the test program.

5

17. A system as recited in claim 16, wherein the test program exercises the application API by calling methods from the plurality of methods comprising the API.

18. A system as recited in claim 16, further comprising logic that designates
10 the application as portable when the percentage is greater than a predetermined bar percentage.

19. A system as recited in claim 16, further comprising logic that designates
the application as non-portable when the percentage is less than the predetermined bar
15 percentage.

20. A system as recited in claim 19, further comprising logic that designates
the application as non-portable when an exception occurs during execution of the test
program.

20